Using Linear Programming to Minimize Interference in Wireless Sensor Networks

A THESIS

SUBMITTED TO THE FACULTY OF

UNIVERSITY OF MINNESOTA

BY

Eric Holec

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Nikolaos Papanikolopoulos

September 2013

Abstract

Interference in wireless sensor networks can have a significant impact on power consumption and throughput. In this paper, we address the problem of finding a network topology that minimizes the maximum interference experienced by any sensor in the network. In the standard interference model, each sensor interferes with every other sensor within its communication range. We approach the problem of minimizing interference by creating a linear relaxation to a similar problem with a different interference model. Using randomized rounding, this relaxation gives an $O(OPT \cdot \log n)$ approximation to this new problem. We then show that this solution is an $O(OPT^2 \cdot \log n)$ approximation to minimize the maximum interference using the standard interference model. If $OPT = O(\log n)$ (as is the case in most networks), this is an improvement over existing best known $O(OPT \cdot \sqrt{n})$ approximation. Additionally, we perform several experiments using simulated sensor networks where our algorithm often significantly outperforms its theoretical bounds.

Table of Contents

	LIST OF FIGURES	**iii**
1	**INTRODUCTION**	**1**
2	**RELATED WORK**	**2**
3	**APPROXIMATION ALGORITHM**	**3**
3.1	**Minimum Interference as a Integer Linear Program**	**3**
3.2	**Relaxing the Integer Constraints**	**6**
3.3	**Compound Interference**	**6**
3.4	**Randomized Rounding**	**7**
3.5	**Bound on Compound Interference**	**9**
3.6	**Implementation Details**	**9**
4	**EXPERIMENTAL RESULTS**	**10**
5	**CONCLUSION AND FUTURE WORK**	**13**
6	**WORKS CITED**	**15**

List of Figures

Figure 1: Examples of the various simulated sensor arrangements used for testing. The different arrangements are uniform distribution (a), normal distribution (b), offset grid (c), and exponential tree (d). 11
Figure 2: The average interference of the linear relaxation (LP) and minimum spanning tree (MST) on the various sensor configurations. On average our algorithm outperformed not only the minimum spanning tree, but also its worst case bounds. 12
Figure 3: The computational cost of our algorithm as the problem size increases. The cost is approximately proportional to n^5. 13

1 Introduction

Wireless sensors networks have many applications. For example, a large number of low cost sensors can be used to monitor water quality, detect forest fires, or provide wide area security. In order to make such systems practical, the individual sensors must be low cost. This limits the capabilities of the sensors. In particular, providing a permanent power source is usually cost prohibitive. Instead, the sensors rely on battery power, making energy usage a significant concern.

Often, sensors can be scaled down in size, reducing their power consumption. Wireless communication, however, does not receive the same benefit. While a simple thermometer may be able to operate on a watch battery for several years, just a few hours of wireless communication would exhaust its power supply. Clearly, minimizing wireless messaging is necessary to extend the life of small wireless sensors.

Given a wireless sensor application, there will be some amount of data transmission required. While reducing this is beneficial, it is highly domain specific and beyond the scope of this paper. Instead, we examine the problem of interference in wireless sensor networks. If two sensors are broadcasting within range of one another, their signals will cause interference. One or more messages will be lost, and therefore need to be retransmitted, resulting in increased power consumption.

We will consider a network to have failed if any of its sensors have failed. This includes failure due to exhausting its power supply. As wireless communication takes a significant amount of power, and interference increases the number of data transmissions needed, it is desirable to find a network configuration where no sensors have a high interference. This is called the minimum maximum interference problem, but for compactness, we will refer to this as simply the minimum interference problem.

The remainder of this paper is divided into four sections. In section 0, we discuss the previous work on the minimum interference problem. Next, section 0 describes our approximation algorithm. We then test our algorithm on several types of simulated networks in section 4. Finally, section 5 summarizes our results and suggests future work.

2 Related Work

The problem of minimizing wireless interference has received a significant amount of attention. A variety of different interference models have been proposed (see [1] for a survey of interference models). In transmitter centric interference models, the concern is over the number of sensors affected by a given sensor's transmissions. These models have been studied in [2] [3]. Benkert, Gudmundsson, Haverkort, and Wolff [4] showed that for many of these types of transmitter centric interference models, a minimum spanning tree solves the problem optimally.

We are more concerned with a receiver centric interference model, as the effects of interference are felt at the receiver, not the transmitter. The work by Moscibroda and Wattenhofer [5] use the same interference model described here, but attempt to minimize the average interference rather than the maximum interference. They present a greedy algorithm that gives an $O(\log n)$ approximation.

For the particular problem of minimizing the maximum interference, Buchin [6] showed that there cannot be a 4/3-aproximation unless $P = NP$. As consequence of their proof, it can also be concluded that even is the interference of the network is a small, finding the minimum interference network is still NP-Hard. If this was not the case, finding a Hamiltonian cycle in a grid graph would also be solvable in polynomial time. Additionally, Bilò and Proietti [7] proved that we cannot do better than a log-factor approximation for several different interference models, including a slightly generalized version of the model we are interested in.

A more constrained version of the maximum interference problem limits the sensors to lie on a line. This is known as the highway model. An algorithm producing $O(\sqrt[4]{n})$ interference network for the highway model was given by Von Rickenbach, Schmid, Wattenhofer, and Zollinger [8].

The work of Halldórsson and Tokuyama [9] generalizes the results of [8] to the case of sensors in a plane, resulting in an $O(OPT \cdot \sqrt{n})$ algorithm. This is the currently best known bound for the maximum interference problem.

3 Approximation Algorithm

First, let us formally define the minimum interference problem:

Let $V = \{v_1, v_2, \dots, v_n\}$ be the positions of a set of sensors in a plane. Let $G(V, E)$ be an undirected graph that spans the sensors. Let $D = \{D_1, D_2, \dots, D_n\}$ be a set of transmission discs where D_i is centered on v_i and the radius of D_i is equal to the distance from v_i to its furthest neighbor in G. The interference of a sensor v_i is the number of discs which cover v_i (not including its own). The minimum interference problem is to find a graph G that minimizes the maximum interference of any sensor.

In the remainder of this section, we will describe an approximation algorithm for the minimum interference problem. First, an integer linear program (ILP) will be given which solves the problem exactly. This ILP will then be relaxed to a linear program. While this relaxation does very poorly, we will give a slight modification which is a relaxation of the related minimum compound interference problem. Next, a randomized rounding scheme is given to convert this relaxed solution back to an integer solution, and the approximation ratio is calculated. Finally, we will show how the solution to the compound interference problem relates to the original interference problem.

3.1 Minimum Interference as a Integer Linear Program

To begin with, we will describe an integer linear program which solves the minimum interference problem exactly. Let A be the $n \times n$ adjacency matrix of the communication graph. For now, let $inter_{max}(A)$ be the maximum interference of any sensor using the communication graph A. Later, we will show how this can be written as a linear objective function, but for now let us add several constraints to ensure that A is a valid communication graph. First, all of the elements of A must be 0 or 1, with the exception of the diagonal, which must always be 0 (sensors cannot connect to themselves). Second, A must be symmetric as we are only considering undirected edges.

We must also ensure that the communication graph spans all of the sensors. In order to make this guarantee, the additional constraint $min - cut \geq 1$ is added. That is, for every $S \subseteq V : \emptyset \neq S \neq V$, there must be at least one edge from S to $V \setminus S$. This is both necessary and sufficient to ensure that the graph is a spanning graph. A naive approach to enforce this would be to add one constraint for each subset requiring the number of the edges leaving the subset to be at least 1. This, however, would result in an exponential number of constraints. Carr et. al. [10] gives an equivalent set of constraints requiring $O(|V|^2)$ inequalities and $O(|V|^3)$ variables. For compactness, we will simple use $min - cut \geq 1$ to refer to the set of constraints and variable needed to enforce connectivity. (1) shows our ILP so far.

$$\begin{array}{llcll} min & inter_{max}(A) & & & \\ s.t. & A_{i,j} & = & \{0,1\} & \forall i \neq j \\ & A_{i,i} & = & 0 & \forall i \\ & A & = & A^T & \\ & min - cut(A) & \geq & 1 & \end{array} \tag{1}$$

Still needed is a linear objective function counting the maximum interference. For any given A, we can construct a matrix R, where $R_{i,j} = 1$ if the communication disc for sensor i interferes with sensor j, and 0 otherwise. Summing the values of any given column j in R gives the total interference for sensor j. The maximum column sum of R is therefore the maximum interference for the given communication graph as show in (2).

$$inter_{max}(A) = max_j \sum_i R_{i,j} \tag{2}$$

Converting (2) to a linear objective function is strait forward. A new variable r_{max} is introduced. Minimizing r_{max} subject to the constraint that r_{max} must be greater than any column sum of R gives us (3).

$$\begin{array}{llcll} min & r_{max} & & & \\ s.t. & r_{max} & \geq & \sum_i R_{i,j} & \forall j \end{array} \tag{3}$$

To ensure that R has the appropriate structure, several constraints are needed. First, $R_{i,j} \geq A_{i,j} : \forall i,j$. This is clearly true as an edge from $i \leftrightarrow j$ in the communication graph means that the communication disc of sensor i must be large enough to cover j, and therefore interfere with it. We must also account for the fact that if there is the edge $i \leftrightarrow j$, every sensor closer to sensor i than sensor j will also be interfered with. Let $k(i,t)$ be the t^{th} furthest sensor from sensor i. We can add another set of constraints, $R_{i,k(i,t)} \geq R_{i,k(i,t+1)} : \forall i, t > 2$. In other words, a sensor that interferes with the $(t+1)^{th}$ furthest sensor from i, must also interfere with the next closest sensor. We start at $t = 2$ as a sensor does not interfere with itself. Combining these constraints with (1) gives (4).

$$\begin{array}{llccl}
min & r_{max} & & & \\
s.t. & r_{max} & \geq & \sum_i R_{i,j} & \forall j \\
& R_{i,j} & \geq & A_{i,j} & \forall i,j \\
& R_{i,k(i,t)} & \geq & R_{i,k(i,t+1)} & \forall i, t > 2 \\
& A_{i,j} & = & \{0,1\} & \forall i \neq j \\
& A_{i,i} & = & 0 & \forall i \\
& A & = & A^T & \\
& min-cut(A) & \geq & 1 &
\end{array} \tag{4}$$

3.2 Relaxing the Integer Constraints

Unfortunately, as (4) is an integer linear program, finding the optimal solution is, in general, NP-Hard. We can easily relax the integer requirement by replacing the constraint $A_{i,j} = \{0,1\}$ with $0 \le A_{i,j} \le 1$. This change, however, only results in the trivial solution of $A_{i,j} = \frac{1}{n-1}\ \forall i,j$. The reason for this will be more obvious when we discuss converting the relaxation back to an integer solution, but for now, let us look at the related problem of finding the minimum compound interference network.

3.3 Compound Interference

The compound interference problem is very similar to the original interference problem, with one important difference. Instead of a sensor having a single communication disc, it has one communication disc for each neighbor.

Let $V = \{v_1, v_2, \dots, v_n\}$ be the positions of a set of sensors in a plane. Let $G = (V, E)$ be an undirected graph that spans the sensors. Let $D = \{D_1, D_2, \dots, D_{2\cdot|E|}\}$ be the set of transmission discs where the two discs $D_{2\cdot s}$ and $D_{2\cdot s+1}$ are centered on the two endpoints of E_s, and have radii equal to the distance between the endpoints. The interference of a sensor v_i is the number of discs which cover v_i (not including its own). The minimum compound interference problem is to find a graph G that minimizes the maximum compound interference of any sensor.

As the compound interference problem is very similar to original interference problem, its linear relaxation needs very few changes. The constraint $R_{i,k(i,t)} \ge R_{i,k(i,t+1)}$ needs to be changed to $R_{i,k(i,t)} \ge R_{i,k(i,t+1)} + A_{i,k(i,t)}$. This is because as you get closer to a sensor with multiple neighbors, the interference will increase as more of that sensor's communication discs cover you. Another change in the relaxation is that the inequalities $R_{i,j} \ge A_{i,j} : \forall i,j$ can be dropped as they are already accounted for by the previous constraint. This leads to (5).

$$
\begin{array}{llcll}
\min & r_{max} & & & \\
s.t. & r_{max} & \geq & \sum_i R_{i,j} & \forall j \\
 & R_{i,k(i,t)} & \geq & R_{i,k(i,t+1)} + A_{i,k(i,t)} & \forall i, t > 2 \\
 & A_{i,j} & \geq & 0 & \forall i \neq j \\
 & A_{i,j} & \leq & 1 & \forall i \neq j \\
 & A_{i,i} & = & 0 & \forall i \\
 & A & = & A^T & \\
 & min - cut(A) & \geq & 1 &
\end{array}
\tag{5}
$$

3.4 Randomized Rounding

Solving (5) no longer produces a trivial answer, but it may still have non-integer values. Converting this relaxed solution back to a valid graph is a simple process. For each edge $i \leftrightarrow j : i < j$, generate a uniformly distributed random number $r_{i,j}$, in the range $(0,1]$. If $A_{i,j} \geq r_{i,j}$, add the edge $i \leftrightarrow j$ to the communication graph. This is repeated a number of times until the graph is connected.

First, we will determine how many rounding iterations are necessary to ensure that the communication graph is connected. Let us look at an arbitrary set of sensors, $S \subset V$. Let E_S be the set of edges between S and $V \setminus S$. We would like to know the probability that at least one edge from E_S is selected in a single iteration. Recall that we constrained our relaxation so that the $min - cut(A) \geq 1$. This means that the total weight of the edges in E_S must be at least 1. In the worst case, the weight of each edge is equal to $\frac{1}{m}$ where $m = |E_S|$. The probability that no edge is selected is $(1 - \frac{1}{m})^m$. For all $m \geq 1$, $(1 - \frac{1}{m})^m \leq \frac{1}{e}$. The probability that at least one edge is selected is therefore at least $1 - \frac{1}{e} \approx 0.63$.

Using the above probability, we know that the expected number of disconnected components in our graph after the first rounding iteration is less than or equal to $n \cdot (1 - \frac{1}{e})$. In fact, after any iteration, we can expect the number of disconnected components to decrease by a factor of $1 - \frac{1}{e}$. The expected number of disconnected components after t iterations is therefore

$n \cdot (1 - \frac{1}{e})^t$. Let us assign $t = c \cdot \log n$. By choosing any constant $c \geq \frac{1}{1 - \log(e-1)}$, the expected number of disconnected components after $c \cdot \log n$ iterations is bounded (in fact, if $c = \frac{1}{1 - \log(e-1)}$, the expected number of disconnected components is exactly 1). Finally, we need to show that a single component can be reached with high probability. Let $comp_t$ be the number of disconnected components after t rounding iterations. Using Markov's inequality, $\Pr(comp_t \geq 4) \leq \frac{E(comp_t)}{4}$. Substituting the expected number of components gives $\Pr(comp_t \geq 4) \leq \frac{1}{4}$, and therefore $\Pr(comp_t < 4) \geq \frac{3}{4}$. With high probability, we will have fewer than 4 disconnected components after our rounding. The graph can be then connected by adding at most 3 more edges to the graph which will increase the interference by at most 3.

Given that we are performing $c \cdot \log n$ iterations of our rounding scheme, we can now calculate the expected interference. This is very strait forward. Recall that we constructed the matrix R, where $R_{i,j}$ is the interference that the sensor i causes to sensor j and that $\sum_i R_{i,j}$ is the total interference at sensor j. In our rounding scheme, we select an edge with probability equal to $A_{i,j}$. Any sensor that would be covered by the communication disc created by that edge will have its interference increased by 1 with probability $A_{i,j}$, and 0 with probability $1 - A_{i,j}$. The edge $i \leftrightarrow j$ therefore increases the expected interference of any sensor covered by its communication disc by $A_{i,j}$. The total expected interference of any sensor is, therefore, the sum of weights of the edges which may cause it interference which is exactly what $\sum_i R_{i,j}$ counts. Effectively, the linear program described in (5) is minimizing the expected compound interference after one rounding iteration. This is no greater than the optimal interference as the solution to (5) is a relaxation of the compound interference problem. Because we perform $c \cdot \log n$ iterations, the expected interference is no more than $OPT_{c\_inter} \cdot c \cdot \log n$, where OPT_{c_inter} is the interference of the optimal compound interference network. We can again make use of Markov's inequality to show that the probability of the interference being less than $4 \cdot OPT_{c\_inter} \cdot c \cdot \log n$ is at least $\frac{3}{4}$.

The probability that the randomized rounding scheme produces a solution that it both connected and no more than a factor of $4 \cdot c \cdot \log n$ from OPT_{c_inter} is at worst $\frac{1}{2}$. This gives an

$O(OPT_{c\_inter} \cdot \log n)$ approximation to the compound interference problem. As there are a polynomial number of constraints and variables in the linear program, and LPs can be solved in polynomial time, the approximation algorithm is also polynomial in its computational complexity.

3.5 Bound on Compound Interference

Ultimately, we would like to conclude that by finding a network with low compound interference, we have also found a good solution for the original interference problem. While the compound interference of an arbitrary network can be significantly more than the standard interference for, we can apply some bounds.

First, we need to understand when the compound interference will be greater than the standard interference. In the compound interference case, when a sensor is connected to several other sensors, instead of having a single communication disc, there is one communication disc for each neighbor. It is clear that the compound interference caused by a sensor can be at most equal to its degree, d_i, as this is the number of communication discs it has. As a direct consequence, the compound interference a sensor i causes another sensor is no more than d_i times the standard interference. This means $c\_inter(A) \leq d_{max} \cdot inter(A)$, where d_{max} is the maximum degree of any sensor in the graph. It is also trivial to see that $d_{max} \leq inter(A)$, as a sensor with a given degree must also have at least that much interference.

Given that for the optimal interference network, there must be a compound interference network with no more than $d_{max} \cdot OPT_{inter}$ interference, we can conclude that $OPT_{c\_inter} \leq d_{max} \cdot OPT_{inter} \leq OPT_{inter}^2$. Additionally, the standard interference of a network will never be greater than its compound interference. This means our approximation algorithm givens an $O(OPT^2 \log n)$ approximation.

3.6 Implementation Details

There are several practical optimizations that can be made to this this algorithm. First, the randomized rounding can be terminated early once the network is connected. The $c \cdot \log n$ iterations are needed in the worst case. In the best case, all of the elements of A are already integer and only one iteration is needed. Second, if an edge is chosen between two sensors that already are connected, that edge does not need to be added to the solution. This change prevents loops and can have a significant impact on the interference.

The algorithm described, while polynomial in complexity, can be rather slow. The most significant factor is from the min-cut constraints which add $O(n^3)$ variables to the linear program. Alternatively, the LP can be solved with no connectivity constraints. After solving the LP, if the min-cut of A is at least one 1, you are done. Otherwise, add the constraint that the total weight of the edges between the two partitions formed by the min-cut must be at least 1, and solve the LP again. It may take a number of iterations before the min-cut at least 1, but in our experiments, we found the number of cuts needed is typically linear in the size of the problem.

4 Experimental Results

In order to test the practical performance of our algorithm, several simulated sensor networks were created with varying properties. In the first data set, the sensors were randomly placed in the unit square using a uniform distribution. Majid et. al. [11] showed that in this case, the minimum spanning tree can be expected to produce an $O(\log n)$ interference network. The second data set was generated by placing sensors randomly using a normal distribution. This could be similar to real world situations such as deploying sensors from an aircraft. The third data set consisted of sensors placed in a regular grid with spacing of one unit. Each sensor was then perturbed randomly by $[-0.25, 0.25]$ in the x and y directions using a uniform distribution. This setup was chosen because it would be similar to how sensor might be placed if maximum coverage was desired. There also exists a constant interference network for this arrangement. The final data set consisted of sensors in an arrangement called an exponential tree. Sensors are located at $(2^i, 0)$, $(2^i, 1.05 \cdot 2^i)$, and $(1.5 \cdot 2^i, 1.75 \cdot 2^i)$ for $i = 1, 2, \dots, n/3$. This situation is not likely to occur in real world applications, but is of interest because this is one case where the minimum spanning tree will produce an $O(n)$ interference network, while the optimal network has $O(1)$ interference. Figure 1 shows examples of the four configurations.

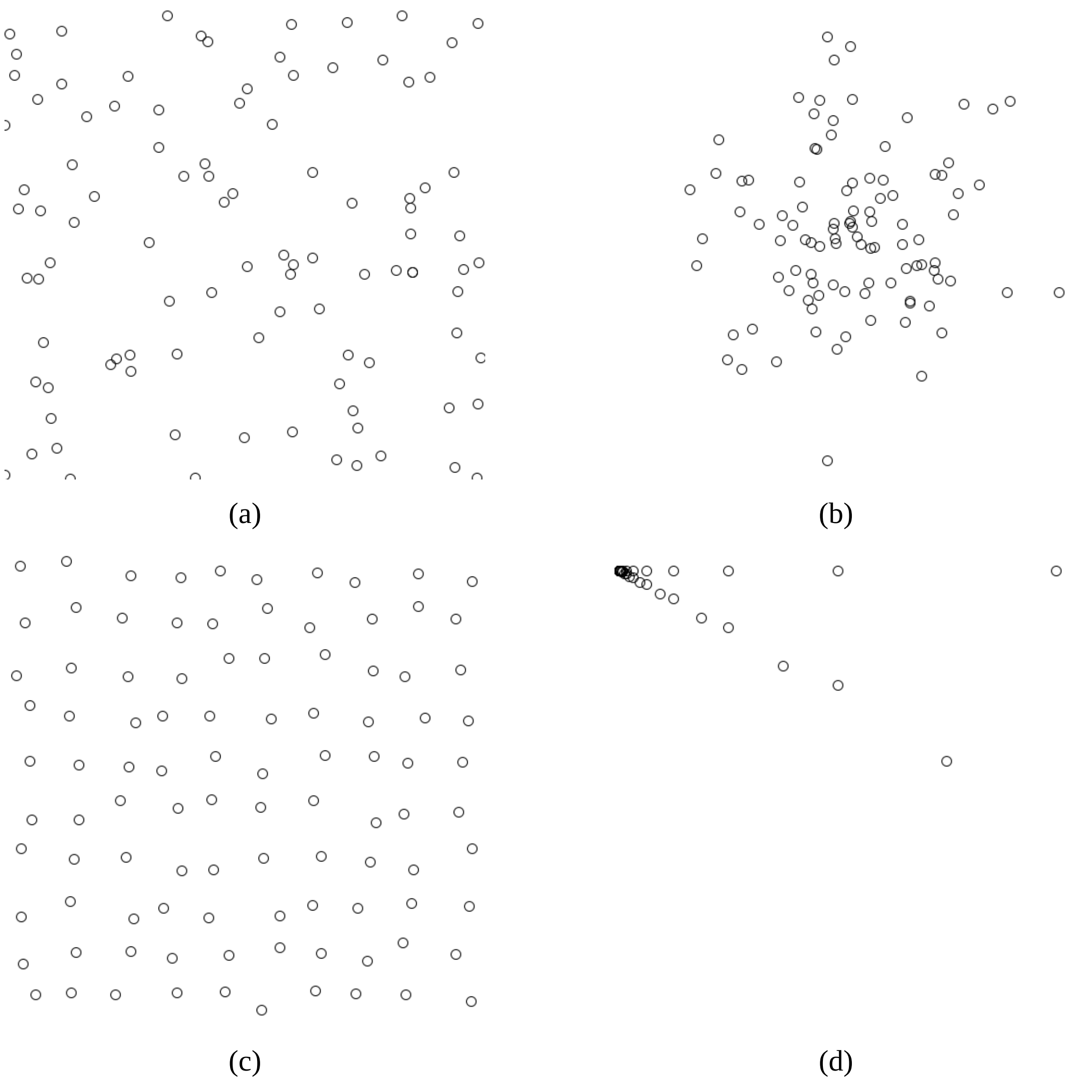
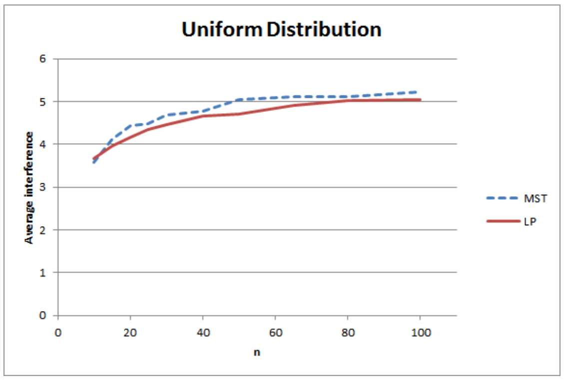
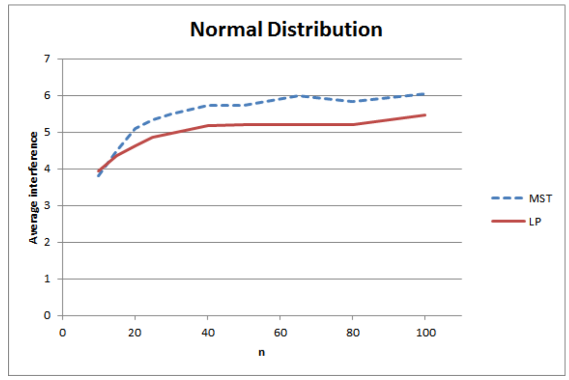


Figure 1: Examples of the various simulated sensor arrangements used for testing. The different arrangements are uniform distribution (a), normal distribution (b), offset grid (c), and exponential tree (d).

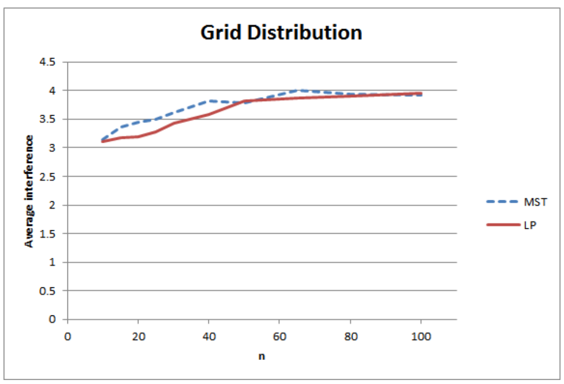
For each of the four network configurations, 100 random networks were generated with sizes $n = \{10,15,20,25,30,40,50,65,80,100\}$. Figure 2 shows the average interference of the network produced by our algorithm as the problem size increases. For comparison, the average interference of the minimum spanning tree is also given. In almost all cases, our algorithm averaged better than the minimum spanning tree. In particular, the linear relaxation gave a constant interference solution for the degenerate case of the exponential tree. It is important to note that while the average interference was an improvement, individual networks sometimes performed worse than the minimum spanning tree on a given sensor arrangement. Taking the better of the two algorithms would further improve performance.



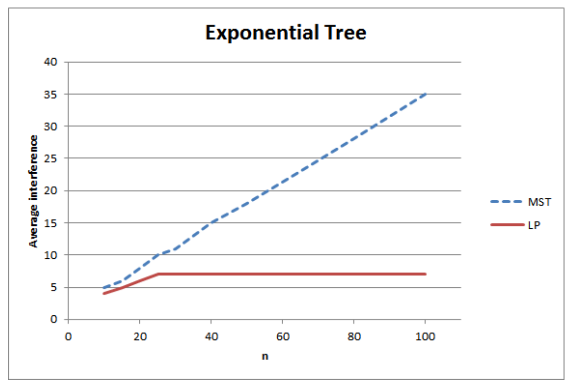
(a)



(b)



(c)



(d)

Figure 2: The average interference of the linear relaxation (LP) and minimum spanning tree (MST) on the various sensor configurations. On average our algorithm outperformed not only the minimum spanning tree, but also its worst case bounds.

Another important concern is the computational cost. Figure 3 shows the computational time as the problem size increases. The cost is approximately proportional to n^5. Almost all of the time was spent solving the linear programs. The solver used was *linprog* from Matlab's Optimization Toolbox [12]. An alternate solver could potentially improve on this. All experiments were carried out on a dual core Intel i5 processor running at 2.6 GHz. The system had 8 GBs of RAM, but the available memory was never a limiting factor.

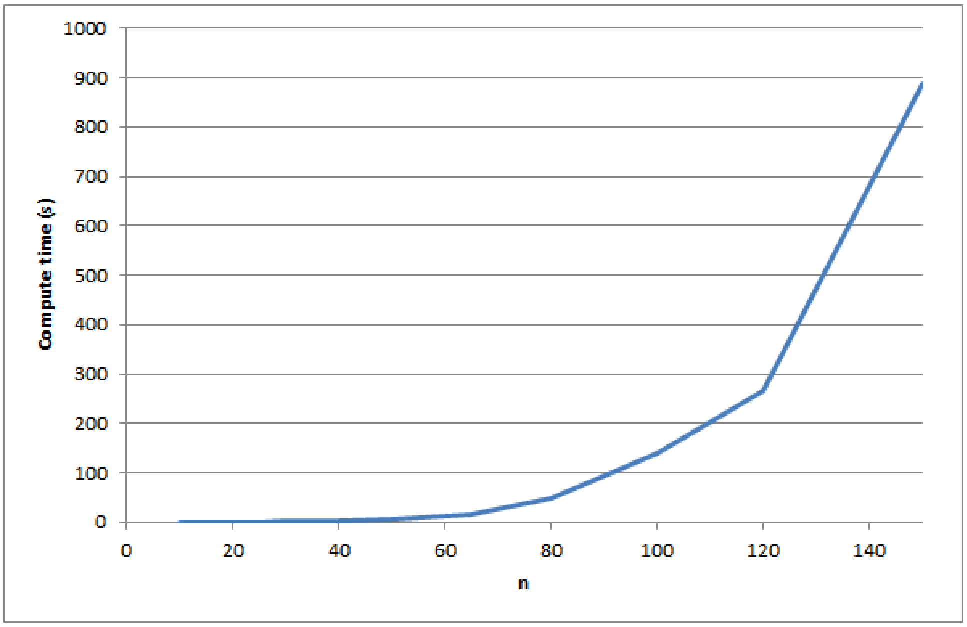


Figure 3: The computational cost of our algorithm as the problem size increases. The cost is approximately proportional to n^5.

5 Conclusion and Future Work

We presented a novel approximation algorithm for the minimum maximum interference problem. By creating a linear relaxation to a similar problem, then rounding that solution back to a valid network, our algorithm produces an $O(OPT^2 \cdot \log n)$ approximation. This improves on the state of the art if $OPT = O(\log n)$, which is the case in most practical situations. Further, we use simulated sensor networks to verify our result and demonstrate that our algorithm often significantly out performs its theoretical bounds.

While we give the worst case bound of $O(OPT^2 \cdot \log n)$, we can provide a slightly tighter bound of $O(d_{max} \cdot OPT \cdot \log n)$. In some sensor configurations, it may be possible to show that the maximum degree of any sensor is significantly less than OPT. In that case, the optimality bounds can be improved. Our experiments also show that our algorithm outperforms these bounds in practice, suggesting that some additional guarantees could be given.

Another area of future work would be in generalizing our algorithm. While we used the same definition of the minimum interference problem as other related work, it does not always accurately represent real world systems. For example, a signal does not immediately stop once it has reached the edge its communication range. The signal gradually deceases in strength, and may still cause some interference. Adapting our relaxation to other interference models would expand its applicability to real world sensor networks.

6 Works Cited

[1] P. Cardieri, "Modeling interference in wireless ad hoc networks," *Communications Surveys & Tutorials, IEEE,* vol. 12, no. 4, pp. 551-572, 2010.

[2] M. Burkhart, P. von Rickenbach, R. Wattenhofer and A. Zollinger, "Does topology control reduce interference?," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, 2004.

[3] K. Moaveni-Nejad and X.-Y. Li, "Low-interference topology control for wireless ad hoc networks," *Ad Hoc \& Sensor Wireless Networks,* vol. 1, no. 1-2, pp. 41-64, 2005.

[4] M. Benkert, J. Gudmundsson, H. Haverkort and A. Wolff, "Constructing interference-minimal networks," in *SOFSEM 2006: Theory and Practice of Computer Science*, Springer, 2006, pp. 166-176.

[5] T. Moscibroda and R. Wattenhofer, "Minimizing interference in ad hoc and sensor networks," in *Proceedings of the 2005 joint workshop on Foundations of mobile computing*, 2005.

[6] K. Buchin, "Minimizing the maximum interference is hard," *arXiv preprint arXiv:0802.2134,* 2008.

[7] D. Bilò and G. Proietti, "On the complexity of minimizing interference in ad-hoc and sensor networks," *Theoretical Computer Science,* vol. 402, no. 1, pp. 43-55, 2008.

[8] P. Von Rickenbach, S. Schmid, R. Wattenhofer and A. Zollinger, "A robust interference model for wireless ad-hoc networks," in *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, 2005.

[9] M. Halldòrsson and T. Tokuyama, "Minimizing interference of a wireless ad-hoc network in

a plane," *Theoretical Computer Science,* vol. 402, no. 1, pp. 29-42, 2008.

[10] R. Carr, G. Konjevod, G. Little, V. Natarajan and O. Parekh, "Compacting cuts: a new linear formulation for minimum cut," *ACM Transactions on Algorithms (TALG),* vol. 5, no. 3, p. 27, 2009.

[11] M. Khabbazian, S. Durocher and A. Haghnegahdar, "Bounding interference in wireless ad hoc networks with nodes in random position," in *Structural Information and Communication Complexity*, Springer, 2012, pp. 85-98.

[12] The MathWorks Inc., "Matlab 2012b," Natick, Massachusetts.